ARC WELDING SEAM TRACKING SYSTEM BASED ON ARTIFICIAL NEURAL NETWORKS

L. Kreft, W. Scheller

Universities of Technology Hamburg-Harburg and Aachen, Germany

A new approach towards controlling an arc welding seam tracking system based on process-oriented data is presented; an artificial neural network is the central controlling element. Special features of neural networks, such as trainability, abstraction capability and fault-tolerance give this controlling system significant advantages. The adaptation of the system to new welding tasks has been simplified substantially in comparison with conventional approaches: Analysing the new process parameters and programming the system is replaced by producing reference data from the new welding process and subsequent automatic learning of an artificial neural network. Results on the basis of a quality function are presented.

INTRODUCTION

Gas-shielded welding is an important method in the metal-working industry for joining metal parts. According to aspects of cost efficiency the single parts are not produced precisely enough to enable welding robots, trained by "teach-in", to follow the real contour. Therefore, seam tracking systems for welding heads gain increasing importance in automated plants. As a result of the precise head positioning those systems are able to compensate the component tolerances and in addition clamping errors and thermal distortions [Dilthey (1), Dilthey and Stein (2)].

Seam tracking systems based on image processing feature severe disadvantages, e. g. unacceptable restrictions for the freedom of head-action by the additional apparatus, stress of the optical elements by welding fume and hot particles, cost extensive control for the very wide variance of luminance conditions as well as data corruption by electromagnetic distortion.

Seam tracking systems based on process-oriented data are much more convenient. No external sensors are required if primary welding parameters –current and/or voltage– are used. For this approach the information about the head position can be extracted easily if welding methods like weaving the arc across the groove or using two electrodes are implemented.

THE APPLICATION CONTEXT

The production of transverse control arms in the car industry is a typical example for a class of welds in the thin sheet sector. As a result of the deep-drawn semifinished products a wide tolerance range for the components must be encountered. Test components are developed for simulating different, typical tolerance conditions. For this application, profiles illustrated in figure 1 had been defined and realised. The width b is reflecting practical bottom sheet changes. For further applications additional test components, shown in figure 2, are defined.

Some work has been done to formulate classically based rules that will result in control information

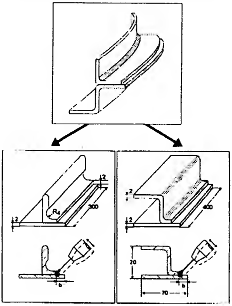

Figure 1: Components for simulation of transverse control arms welding

'Intelligent Systems Engineering', 5–9 September 1994, Conference Publication No. 395, © IEE, 1994

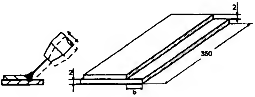

Figure 2: Additional component for simulation

even from signals that suffer from disturbing influences [Eichhom et al (3,4)]. As the results of those efforts were not satisfactory, alternative methods had to be examined. The approach to a solution of this problem as presented here involves the use of a trainable system in the form of an artificial neural network which can be trained to respond to the electrical signals of the welding process.

PATTERN RECOGNITION BY ARTIFICIAL NEURAL NETWORKS

In recent years artificial neural networks have been the subject of research in the field of science, medicine and technology to an increasing extent. Historically, the roots of artificial neural networks lie in approximation theory. Otherwise neurobiology developed simulation models which imitate signal processing of the brain in a simplified form to investigate human learning behaviour [Wassermann (5), Hecht-Nielsen (6)]. From the technical point of view there are some special properties worth mentioning, such as trainability, abstraction capability and fault-tolerance in pattern recognition. Therefore, artificial models of neural networks are used in industrial fields that involve pattern recognition applications, for instance for speech generation and recognition, identification of registration plate numbers on cars or the control of self-orienting vehicles [Sejnowski and Rosenberg (7), Rojaz (8), Raus et al (9)].

In the case of pattern recognition, as done in the described system, normally multi-layer perceptrons are used. This kind of network has the advantage of no closed loops, thus simplifying the formal development. Typical applications are using one or two hidden layers. The information flows in one direction, i.e. each computing cell (artificial neuron) transmits its result to the next layer. The behaviour of such a network will only be defined by the weights fixed in each cell.

The problem is to adjust all the weights of the artificial network so that the recognition or classification will be done correctly; this is done by learning in contrast to the programming classical computers. In our realisation we use supervised learning based on the backpropagation-algorithm. This method assumes a supervisor that knows different input vectors as well as the corresponding output vectors. In the learning phase the supervisor is presenting input vectors to the artificial network. The network itself will produce an output vector depending on the actual weights. As the supervisor knows the correct output vector he can compute the error and change the weights so that the error of the output vector will be minimised for all patterns of the test-sample. This has to be done iteratively until the result will satisfy the user's claim.

The most known algorithm for technical applications using multilayer-perceptrons is the backpropagation-algorithm. It is based on the Fermi-function $F_{(0)} = 1/(1+e^{-x})$ for the threshold-function in each cell, because its derivatives exist. So the modifications for the weights can be derived from the error at the output layer propagating back to the input layer.

$$\Delta w_{ij}{}^{t} = \alpha \cdot \varepsilon_i^t \cdot y_j^t \cdot y_i^t \cdot \left(1 - y_i^t\right) + m \cdot \Delta w_{ij}{}^{t-1}$$

$$\Delta w_{jk}{}^{t} = \alpha \cdot \sum_i \varepsilon_i^t \cdot y_k^t \cdot y_j^t \cdot \left(1 - y_j^t\right) \cdot w_{ij}^t \cdot y_i^t \cdot \left(1 - y_i^t\right) + m \cdot \Delta w_{jk}{}^{t-1}$$

- y: neuron output
- ε_i: error of the ith output
- i: output layer
- j: last hidden layer
- k: second hidden layer from the back
- t: time step

The learning factor $\alpha \in (0,1)$ will influence the learning speed as well as the stability. The momentum $m \in (0,1)$ will consider preceding weight modifications and thus stabilise the procedure like a low pass filter. By repeating this operation in an iterative process with a number of characteristic input and output signals, at the end of the training "course" an artificial neural network has learned to behave in a manner which is inconceivable for conventionally programmed computing machines. The abstraction capability generated by the training and the associated robustness in pattern recognition allow the network to classify signal patterns which were not included in the training session with a high degree of certainty. Thus, when for example identifying hand-written characters, it is possible to recognise letters or digits written in unknown handwriting and to classify these correctly, too. [Ritter et al (10), Eckmiller (11), Schöneburg et al (12), (5) , (6), Hubbard and Jackel (13)].

The welding head seam tracking system described here makes use of these properties by replacing the pattern classification algorithm previously used by a neural network. Whereas the user of a conventionally implemented system had to specify the rules according to which the system records the influence on the electrical process signals of varying geometry of the parts to be joined, this now all takes place in an automatic learning operation with a typical sample of process data.

ARTIFICIAL NEURAL NETWORKS FOR WELDING APPLICATIONS

In the presented application the actual position of the welding head should be obtained from the electrical process signals. Those signals are distorted by normal electrical noise as well as from instabilities of the arc welding process. Especially these instabilities lead to the use of an artificial neural network for controlling the seam tracking system. The system structure is shown in figure 3. First of all the neural part of the controller will be discussed.

The realisation of the artificial neural network implies two major steps: the layout of the network and the definition of the learning procedure.

The input layer of the system was adjusted to 100 cells, the number of test samples for one sweep. Different approaches using 200 or 300 input cells for considering 2 or 3 adjacent sweeps did not result in significantly better performance; the much higher extent had not been justified.

The output layer contained 3 to 9 cells depending on the groove geometry and other process parameters like torch-groove distance, or wire-feed velocity. In several tests it could be verified that too many output cells will result in untrainable systems; in this case it had been tried to train the network for information that is not available from the presented data and the trained network will respond to insignificant statistic data. In this application the corresponding limit had to be found experimentally. Otherwise, from the control unit's point of view a high resolution for the predicted position is desired so that a high number of output cells should be claimed. In other industrial applications of artificial neural networks this aspect is no problem, when the set of embedded information is given by the application, i. e. letter classification.

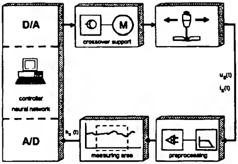

Figure 3: The neural-controlled seam-tracking system for the welding head

For the amount of cells in the hidden layer(s) no general recommendation can be given. In the presented application a number of 70 cells experimentally proved to be enough for one hidden layer; neither more than one layer nor more than 70 cells led to better results. As the artificial neural network had been simulated on standard PC's (sequential processing of all neural cells) it was important to reduce the number of cells to a minimum in order to implement online-capability.

The necessary number of the hidden cells was strongly dependent on the number of the presented test samples.

These representative test samples of process-data combined with corresponding information about the head position had to be produced for the learning process. Therefore, process signals originating from reproducible conditions such as torch position and groove geometry are generated. For this purpose, so-called "reference welds" were performed on the simulation components shown in figures 1 and 2. The torch was applied at an angle of 45° crossways the direction of welding and forced to oscillate mechanically with a frequency of 8 Hz. The torch and oscillating unit were mounted on a three-axis welding positioner. The welding current source used was a transistor source with a longitudinally controlled power pack characterised by low ripple of the welding current and short reaction times to process interference. Both the horizontal torch position and the width b of the projecting bottom sheet were varied in defined steps during the tests.

As an example, figure 4 shows three ground sections with one correct and two off-centre torch positions. An analogue pre-processing electronic unit calculates the linear combination from welding current and voltage before digitalisation and storing. Each welding sweep results in 100 time-dis-

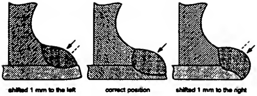

Figure 4: Schematic view of three ground sections

crete scanning values. There are two approaches for using the reference data: unchanged or mean values of some sample-runs under equivalent geometrical conditions (noise filtering). Both ways had been tested; the filtered data resulted in faster learning.

RESULTS

The implemented neural control system had been trained with training-data produced from the different test components under various parameter combinations (figures 1, 2). The amount of samples ranged from 9 (filtered mean values) to 180 for the different simulation components, respectively. After a fixed amount of loops (20-30) the training algorithm changed the sequence of samples within the training-set statistically (1 loop is the presentation of the whole sample-set); otherwise the network would have been trained to sequence-specific information of the data, too.

The learning process is illustrated in figures 5 and 6. The total error as a function of the learning loops is shown in figure 5. Up to approx. 170 loops, the total error is decreasing; after that point it tends to increase again. Therefore the learning factor α had been reduced from 0.7 at the beginning to 0.35 (↓ in figure 5), thus allowing the system to leave local minima when starting the learning process and to stabilise the process to the end. The effect of the momentum is demonstrated in figure 6: Two learning processes with $m = 0$ and $m = 0.7$ are recorded; obviously the training with the momentum of 0.7 converges much faster. So $m = 0.7$ had been chosen for further training procedures.

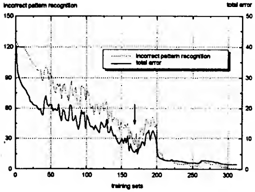

Figure 5: Total error as a function of the training loops

The training procedure will be stopped if one of the following criteria is met: the total error becomes less than a fixed limit, a maximal number of loops has been calculated, or a tolerance condition is satisfied. The tolerance condition is the maximal relative error that any pattern at the output is allowed to have. Typical values in industrial applications are 5-30%; in this application 20% has been chosen. The tolerance condition will prevent the network from overlearning, where the network begins to adapt to fine-structured information in the training set, such as noise; this will reduce the capability of generalisation and fault-tolerance.

For evaluating the artificial neural network during the recall-phase (application-phase) untrained reference data from the test components with known parameters are classified by the network. Differences between the network and the expected responses could be calculated and evaluated. Two quality functions have been introduced. The first Q_{TC} gives the percentage of correct classified patterns, where a classification is correct it complies with the tolerance condition (TC). The second Q_{WTA} is given by

$$Q_{WTA} = \frac{1}{P_{tot}} \cdot \sum_{i=1}^{P_{tot}} k(i) \cdot P(i)$$

P: classified pattern

P_{tot}: total number of tested patterns

k: classification factor, $k \in \{-1, -0.5, 0, +0.5, +1.0\}$

This quality-function is based on a discrete classification (Winner Takes All) and is weighted con-

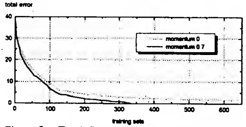

Figure 6: The influence of the momentum on the training process

TABLE 1: Example for computing Q_{WTA} (50 tested patterns)

position	1.0 mm to the left	0.5 mm to the left	middle	0.5 mm to the right	1.0 mm to the right
k	0	+1	0	-0.5	-1
quantity	6	30	10	2	2
k•P	0	+30	0	-1	-2

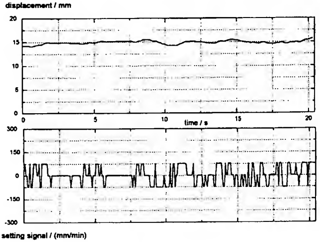

Figure 7: Support position and network-induced setting

trol-specifically; e. g. the class "0.5 mm to the right" is worse than "middle position", if the class "0.5 mm to the left" (grey in table 1) would have been the correct one. In the example of table 1 the quality function is 0.54.

For a real-time application of the neural-controlled robot a program-specific optimisation was necessary to reduce the computing time; so the result of the classification of the process data within one oscillation period is available for the corrective biasing of the welding manipulator. For the evaluation of the network output various strategies were developed. The Winner-Takes-All-method is the most simple method of evaluating the classification. This method suppresses all outputs of the network, except the one featuring the maximum value. It is easier to tune the classificator to its controller tasks by means of evaluating all outputs of the network with the help of linear factors. In order to suppress oscillation, the application of a digital PI-controller is optional. In addition, it is possible to assess the mean of the input values sliding from 2 up to 5 oscillating periods.

The welds performed on the specimen components produced convincing results: Over a seam length of 300 mm lateral position deviations of up to 30 mm were corrected successfully (welding speed: 1000 mm/min, welding current: 200 A, welding voltage: 21,5 V, gas mixture: Ar 90%, CO_2 10%). The width of the bottom sheet varied between 1 mm and 5 mm. Welds of parallel positioned components were also carried out without any problems. Figure 7 shows the progression over the time of the crossover support and the network-induced setting signal when welding a component according to figure 1, right below.

Here, the Winner-Takes-All-strategy was applied, the PI-controller was out of action. The system responded quickly enough to manual manipulations of the torch-position during welding. Critically low widths of the bottom sheet (b < 2 mm) were classified correctly, whereas the relative position of the welding head was slightly corrected in the direction of the upper sheet. This behaviour is naturally an advantage as far as the welding technology is concerned; a surprising result, especially as the network had not been trained for this situation separately. The neural welding head seam tracking system showed exceptionally robust control behaviour which was not sensitive to interference.

CONCLUSION

The experiences described herein showed that artificial neural networks represent an effective evaluation strategy and an efficient means of signal processing in process-oriented welding seam tracking systems. In this connection, the substantial properties of learning by training on the basis of examples, the abstraction capability and the fault-tolerance in respect to process-interference are particularly advantageous. The experimental data proved that the development of an universal welding seam tracking system based on artificial neural networks is applicable to a wide range of welding tasks. The new concept works in connection with analogue electronics which are necessary in conventional systems, and the reformulation of rules in a conventional computer-based welding seam tracking system can be dispensed by the less costly training of a neural network. The low expenditure required when changing to a new welding task offers economic advantages for the use of neural welding seam tracking systems, especially for the production of small and medium-sized batches. Further systematic investigations to define possible fields of application, to assess performance boundaries and to define adapted hardware components based on FPGA's for performance-optimised neural networks are planned.

LITERATURE

1. Dilthey U., 1988, Sensortechnik für Schweißroboter: Entwicklungsstand und Tendenzen, Bänder Bleche Rohre, 7, 32-35

2. Dilthey U. and Stein L., 1991, Industrial Robot Systems for Electrical Arc Welding, IIW-Document XII-1254

3. Eichhorn E., Borowka J. and Habedank G., 1989, Prozeßorientierte Schweißkopfführungssysteme für das automatische MAG-Schweissen von Dünnblechen, DVS-Bericht 89, DVS-Verlag, Düsseldorf, 88-101

4. Eichhorn E., Borowka J. and Habedank G., 1989, Sensorgeführtes Roboterschweißen dünner Bleche - ein neues prozeßorientiertes Schweißkopfführungssystem zur Erkennung von Toleranzen der Fügestellengeometrie, DVS-Bericht 118, DVS-Verlag, Düsseldorf, 40-42

5. Wassermann P., 1989, Neural Computing, Van Nostrand Reinhold, New York

6. Hecht-Nielsen R., 1990, Neurocomputing, Addison-Wesley Publishing Company, Bonn Amsterdam New York

7. Sejnowski T. and Rosenberg C. R., 1986, NETtalk: A parallel that learns to read aloud, Technical Report JHU/EECS-86/01, John Hopkins Univ., Baltimore

8 Rojaz R., 1993, Theorie der neuronalen Netze, Springer-Verlag, Berlin Heidelberg New York

9. Raus M., Kreft L. and Ameling W., 1991, Ein System zur automatischen Kennzeichenerkennung aus Videobildern mit Hilfe neuronaler Verfahren, 13. DAGM-Symposium Mustererkennung, Munich

10. Ritter H., Martinez T. and Schulten K., 1990, Neuronale Netze, Addison-Wesley Publishing Company, Bonn Amsterdam New York

11. Eckmiller R., 1990, Advanced Neural Computers, North Holland, Amsterdam New York Oxford Tokyo

12. Schöneburg E., Hansen N. and Gawelczyk A., 1990, Neuronale Netzwerke, Markt&Technik Verlag, Munich

13. Hubbard W. and Jackel L., 1990, Handwritten Digital Recognition with a Backpropagation Network, Neural Information Processing Systems, Morgan Kaufmann, 2

Weld Modeling and Control Using Artificial Neural Networks

George E. Cook
Robert J. Barnett
Kristinn Andersen
Alvin M. Strauss

Vanderbilt University
Engineering School
Box 1826, Station B
Nashville, TN 37235

Abstract - Artificial neural networks were evaluated for monitoring and control of the Variable Polarity Plasma Arc Welding process. Three areas of welding application were investigated: weld process modeling, weld process control, and weld bead profile analysis for quality control.

I. Introduction

The arc welding processes play an important role in modern manufacturing. Despite the widespread use of arc welding for joining metals, controlling most welding processes still requires considerable skills and experience on behalf of the human welder. Total automation of welding has not yet been achieved, largely because the physics which determine the success of any welding task, are not yet fully understood and quantified.

Arc welding is categorized into a number of processes [1], each of which is characterized by the nature of the applied welding methodology and the techniques required to carry it out. Common to all of the processes is the use of an electric arc which provides heat to the welded joint. The purpose of welding is to apply this heat to melt the base metal, and the optional filler metal, and thus bring about coalescence of the joined metals. There are a number of variables, controllable to varying degrees, which affect the outcome of the welding process. The arc heat, applied to the metal, is to a large extent determined by the arc voltage and the current. Both of these can be well defined and controlled [2-4]. The overall arc efficiency has a very significant effect on the heat as well, but it is neither well known nor controlled. The travel speed of the electrode, as it moves along the welded joint, is a major factor in determining the overall characteristics of the weld as well. Other factors, such as the properties of the welded materials, are of principal importance, too, in obtaining the desired weld qualities.

As for the properties of the weld itself, the overall size of the molten weld bead is one of major importance. Specifically, certain geometric dimensions, such as crown width, root width or penetration are among the parameters which should be controlled throughout the process. Other parameters, which may be to a lesser extent quantified, such as overall weld appearance and lack of discontinuities, should ultimately be controllable as well.

The main objective of this research was to explore applications for artificial neural networks in arc welding control, and in particular the Variable Polarity Plasma Arc Welding (VPPAW) process. The VPPA welding (W) technique integrates two features, a variable polarity (VP) and a plasma arc (PA).

Plasma arc welding of aluminum has been extensively used and developed at NASA's Marshall Space Flight Center using square wave ac with variable polarity (VPPA) [5]. With the variable polarity process, oxide removal before welding is not required for most aluminum alloys. NASA and its subcontractors use the VPPAW process in the space shuttle external tank fabrication.

The plasma arc differs from the ordinary unconstricted welding arc in that, while ordinary welding arc jet velocities range from 80 to 150 meters per second, plasma arc jet velocities range 300 to 2000 meters per second in welding usage [6]. The high plasma arc jet velocities are produced by heating the plasma gas as it passes through a constricting orifice. In VPPA welding of certain ranges of metal thicknesses, appropriate combinations of plasma gas flow, arc current, and weld travel speed will produce a relatively small weld pool with a hole (called a *keyhole*) penetrating completely through the base metal.

VPPAW process modeling was investigated to determine if attributes of the weld bead, such as the crown width, root width or penetration could be predicted based on the corresponding equipment parameters (arc current, torch standoff, travel speed, etc.) Emphasis was placed on direct control of the weld bead geometries, where the user could specify the desired crown and root widths, and the proposed control system would select and maintain the equipment parameters necessary to achieve the desired results. Various neural net-

0-7803-1462-x/93$03.00 ©1993IEEE

work methodologies were investigated for the purpose of arc welding modeling and control.

Additionally, artificial neural networks were used to analyze digitized weld bead profiles, obtained from a laser-based bead surface scanner presently used at NASA. The networks provided an improved method for automatically detecting the location of significant bead attributes, such as the crown, undercuts, and edges.

II. Arc Welding As A Multivariable Process

Any arc welding process is controlled by a number of parameters, and the ultimate objectives of the process are specified in terms of numerous parameters as well (refer to Fig. 1). As a result, any arc welding process can generally be viewed as a multiple-input/multiple-output system. The lack of reliable, general, and yet computationally fast, physical models of this multivariable system makes the design of a generalized real-time controller for arc welding nontrivial. Each process has its own characteristics which usually are related to a number of external parameters. Generally, these relationships are not well quantified.

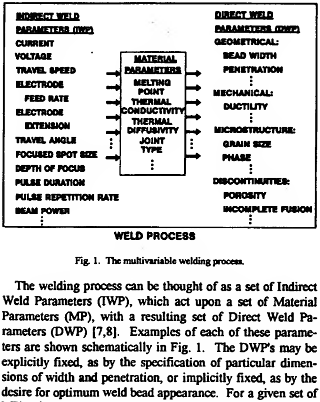

Fig. 1. The multivariable welding process.

The welding process can be thought of as a set of Indirect Weld Parameters (IWP), which act upon a set of Material Parameters (MP), with a resulting set of Direct Weld Parameters (DWP) [7,8]. Examples of each of these parameters are shown schematically in Fig. 1. The DWP's may be explicitly fixed, as by the specification of particular dimensions of width and penetration, or implicitly fixed, as by the desire for optimum weld bead appearance. For a given set of MP's, the IWP's must be chosen appropriately. Conceptually, the IWP's may be thought of as the "causes", with the DWP's being the "effects". In terms of the multivariable weld process control problem, the function of the control system is to maintain the DWP's at a constant state despite changes in the welding environment (primarily changes in MP's), by making appropriate changes in the IWP's. Alternatively, the control system must bring about desired changes in the DWP's while operating in a region of constant MP's by appropriate changes in the IWP's.

One of the basic issues to be considered regarding the generalized control for arc welding is to determine what is to be controlled and which parameters are accessible to enact control actions on the process. In other words, the input parameters and the output parameters of the process, or the plant, have to be determined. The input parameters to any arc welding process typically include those controlled by the welding equipment, such as arc current, voltage, torch travel speed and filler wire rate. Most of these equipment parameters may be adjusted on-line and thus they are available for real-time control of the welding process. Other input parameters may be selected by the welder, but they are usually static and not changeable during welding. These parameters include the electrode type and dimensions, filler wire composition and dimensions, workpiece dimensions and compositions, etc.

III. Background: Artificial Neural Networks

Artificial neural networks (ANN) have gained prominence recently among researchers of nonlinear systems. As the name implies, these networks are computer models of the processes and mechanisms that constitute biological nerve systems, to the extent that they are understood by researchers.

In addition to neural networks' usefulness in solving complex nonlinear problems, they are attractive in view of their high execution speed and their relatively modest computer hardware requirements. Beyond the variety of personal computer and workstation implementations that are available commercially, several efforts are presently underway to take advantage of the inherent parallelism of neural networks. The calculations taking place within the networks are not generally carried out in a serial fashion since they can be conducted in parallel. Although present computers implement neural networks very efficiently, based on the traditional, serial, Von-Neuman architecture, emerging parallel computers will be ideally suited for neural network implementations.

A. The Back propagation Network

The back propagation neural network [9] was used as the basic structure for the applications discussed here. A neural network and its adaptation procedure using the back propagation algorithm will be illustrated by an example.

Fig. 2 shows a small neural network consisting of 8 *nodes*, arranged in two *hidden layers* of 3 nodes each and one output layer of 2 nodes. Each node (or *processing element*)

resembles the connected neurons in biological systems (refer to Fig. 3). A processing element accepts one or more signals, which may be produced by other processing elements or applied externally (e.g., provided by a process sensor). The various signals are individually amplified, or weighted, and then summed together within the processing element. The resulting sum is applied to a specific *transfer function*, and the function value becomes the output of the processing element.

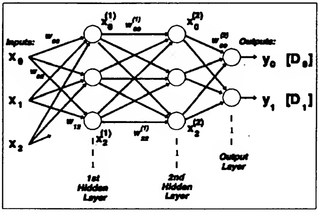

Fig. 2. A back propagation neural network.

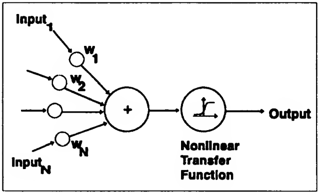

Fig. 3. The processing element of a neural network.

Referring to Fig. 2, each node *i* in the first hidden layer produces a single numeric output which we denote as $x_i^{(1)}$. Similarly the nodes of the second hidden layer are labeled $x_0^{(2)}$ through $x_2^{(2)}$. The 3 inputs and 2 outputs of the network are x_0 through x_2 and y_0 through y_1 respectively. Each node accepts numeric data through a number of *input links*, each of which multiplies the input data with a *weight* factor. The weight factor associated with the link from $x_i^{(1)}$ to the node producing $x_j^{(2)}$ is annotated as $w_{ij}^{(1)}$ and a similar convention holds for the links between other layers. Each node calculates its output by summing its weighted inputs and using the result *s* as the argument of a nonlinear function associated with the node. For our application this function is the same for all nodes:

$$f(s) = [1 + \exp[-(s-c)]]^{-1} \quad (1)$$

where *s* is the sum of the node inputs and *c* is an internal offset value. Clearly the node output will be confined to the range $0 < f(s) < 1$. Because the limiting values, 0 and 1, will only be approached as s approaches +/- infinity, all input and output data are scaled so that they are confined to a subinterval of [0...1]. A practical region for the data is chosen to be [0.1...0.9]. In this case each input or output parameter p is normalized as p_n before being applied to the neural network according to:

$$p_n = [(0.9 - 0.1)/(p_{max} - p_{min})](p - p_{min}) + 0.1 \quad (2)$$

where p_{max} and p_{min} are the maximum and minimum values, respectively, of data parameter p. The network starts calculating its output values by passing the weighted inputs to the nodes in the first layer. The resulting node outputs of that layer are passed on, through a new set of weights, to the second layer, and so on until the nodes of the output layer compute the final outputs.

Before practical application, the network has to be *trained* to perform the mapping of the three input parameters to the two output parameters. This is done by repeatedly applying training data to its inputs, calculating the corresponding outputs by the network, comparing them to the desired outputs, and altering the internal parameters of the network for the next round. The training starts by assigning small random values to all weights (w_{ij}) and node offsets (c_j) in the network. The first three input data values are presented to the network which in turn calculates the two output values. Because the initial weights and node offsets are random these values will generally be quite different from the desired output values, D_0 and D_1. Therefore the differences between the desired and calculated outputs have to be utilized to dictate improved network values, tuning each weight and offset parameter through back propagation. The weights preceding each output node are updated according to

$$w_{ij}(t + 1) = w_{ij}(t) + nd_j x_i^{(2)} \quad (3)$$

where *n* is a *correction gain* and d_j is the *correction factor*

$$d_j = y_j(1 - y_j)(d_j - y_j) \quad (4)$$

Clearly, each weight will be increased if the calculated output from its node is less than the desired value, and vice versa. The correction factors used to update weights preceding hidden layer nodes are updated according to

$$d_j = x_j \cdot (1 - y_j) \sum_k (d_k \cdot w_{jk}) \quad (5)$$

where the k applies to the node layer succeeding the one currently being updated. The offset parameter c of each node is treated as an additional weight factor and updated in the same manner.

The weights and offsets of the neural network are recalculated during the back propagation as outlined above. Then the network repeats calculation of output values based on the same input data, compares them to the desired output values, and readjusts the network parameters through yet another back propagation phase. This cycle is repeated until the calculated outputs have converged sufficiently close to the desired outputs or an iteration limit has been reached. Once the neural network has been tuned to the first set of input/output data, additional data sets can be used for further training in the same way. To ensure concurrent network adaptation to all sets of data the entire training process may be repeated until all data transformations are adequately modeled by the network. This requires, of course, that all the data sets were obtained from the same process and therefore the underlying input/output transformation is consistent.

As noted above, the training iteration process may be terminated either by a convergence limit or simply by limiting the total number of iterations. In the former case we use an error measure *e* defined as follows:

$$e = \max_{k=1...K} \left\{ \sum_{m=0}^{M-1} (d_{k,m} - y_{k,m})^2 \right\} \quad (6)$$

where K is the number of input/output data sets used for training, M is the number of network output parameters in each data set, and $(d_{k,m} - y_{k,m})$ is the error in the network calculation of parameter m in data set k. The error measure, e, changes after each round of network weight adjustments. In the long run e decreases as the network is refined by training iterations. Using this indicator one can program the network to terminate the iteration tuning process as soon as e reaches some threshold value, e_0. Alternatively, a given network may not be able to reduce the error measure down to the specified e_0. In that case the iterations may be terminated by simply specifying a maximum number for them.

The *training mode*, as described above, is a precondition for actually applying the neural network in the *application mode*. In this mode entirely new input data is presented to the network which, in turn, predicts new outputs based on the transfer characteristics learned during the training. If this new data is obtained from the same local region of operation of the process as during the training phase, data from the input/output relations should be governed by the same underlying process and the neural network should perform adequately. The neural network is not updated in the application mode.

B. *Neural Network Hardware and Software Speed Considerations*

A number of hardware accelerators and dedicated neural network computers are presently in the design phase or prototype production stage. Furthermore, a number of neural network software vendors have optimized their code for generic numerical accelerator boards or chips [10]. A typical numerical accelerator board may be expected to increase the execution rate of the neural networks by a factor of approximately an order of magnitude. Using such a board, typical execution times for a 15-node network with 2 inputs and 4 outputs is less than 1 millisecond. For a real-time application this would result in an update rate of over 1000 Hz.

The training speed of neural networks is generally very low. During the training process the internal parameters of the network are iteratively adjusted until the network training has been optimized. For the back propagation network the network weights are the adjustable parameters. To illustrate the computations involved, the VPPAW equipment parameter selecting network, discussed in a later section, may be considered. This network had 2 inputs, one hidden layer of 10 nodes, and an output layer of 4 nodes (one node for each of the 4 outputs). The total number of weights in the network is 60. During each training iteration the network calculates output values for each training data set and accumulates the root-mean-square (RMS) differences of the calculated outputs and the desired outputs. Based on the total RMS difference from one iteration, all of the 60 weights are adjusted using the back propagation algorithm. For 10,000 training iterations this requires a total 600,000 applications of the back propagation algorithm. On a 16 MHz 80386-SX computer this takes on the order of 2 hours with software containing rather extensive graphical and debugging overhead. This figure could be improved substantially with more optimized software. Fortunately, the training process is generally not required for the real-time operations. The training is usually carried out off-line and optimized before the network is finally applied in the real-time process.

The network execution time is the critical factor for real-time applications. The key difference between the execution and training times is that execution is not based on iterative trial-and-error calculations, as the training process, and therefore the execution rate is generally much faster than the training rate. Taking the same example as before, the equipment parameter selection network, the measured execution rate on the 80386-SX computer is slightly above 22 Hz. In other words, the network calculates all of its 4 outputs in less than 50 msec. Again, this is achieved with software that is loaded with debugging facilities, graphics, and other features that impair speed. Furthermore, the computer

does not utilize an accelerator or coprocessor board. Based on this, it can be concluded that the neural network approach offers more than ample execution rate for real-time arc welding applications where a 10 Hz update rate is normally sufficient [11].

IV. VPPAW PROCESS MODELING

Data used for the VPPAW process modeling was obtained from the NASA Marshall Space Flight Center [10]. The general weld conditions and setup parameters are summarized in Table I.

TABLE I
GENERAL WELD CONDITIONS AND SETUP FOR NASA VPPAW EXPERIMENT

Date of Experiment	July 31, 1991
Tool	Station #2
Weld Orientation	Vertical
Weld Pass	Root (Keyhole)
Joint Type	Bead-On-Plate
Joint Gap	N/A
Plate Material	2219 Aluminum
Plate Thickness, mm	6.35
Wire Material	No Wire
Ambient Temperature, deg. C	24
Electrode Material	Tungsten
Electrode Diameter, mm	3.96
Electrode Length, mm	69.6
Orifice Diameter, mm	3.18
Electrode Setback (electrode-to-orifice distance), mm	1.12
Orifice Thickness, mm	3.28
Electrode Cooling Water Flow Rate	N/A
Plasma Gas	Argon
Plasma Gas Flow Rate, scfh	6.0
Plasma Gas Pressure, psi	N/A
Shield Gas	Helium
Shield Gas Flow Rate, scfh	100.0
Shield Gas Pressure, psi	N/A
AVC (On/Off)	Off
Initial Torch Standoff, mm	4.0
Initial Straight (Forward) Current, Amps	130.0
Initial Reverse Current, Amps	130.0
Additional Reverse Current, Amps	60.0
Forward Current Time, msec	19.0
Reverse Current Time, msec	4.0
Pilot Current, Amps	25.0
Travel Speed, mm per sec	3.39
Wire Feed Rate, mm per sec	0.0

The VPPAW data used to train and test back propagation networks for weld modeling and equipment parameter selection is given in Table II. Out of the 13 available weld data sets, 3 were not used in the training process, but reserved for testing. These are typed in boldface in the Table.

A back propagation neural network was constructed to model the welding data (refer to Fig. 4). The network used a single hidden layer of 10 nodes, in addition to the 2-node output layer. The input parameters to the network were the torch standoff, forward current, reverse current, and torch travel speed. The model outputs were the weld crown width and the root width. The network was trained for 10,000 iterations. Further training did not improve the modeling performance of the network, and it turned out that with excessive training (e.g., through 50,000 iterations and beyond) the network converged to "memorization" of the training data, rather than capturing the generalities of the process. The trend to memorize, rather than learn, when excessive training is applied is a well known characteristic of back propagation and it was observed in most experiments carried out for this project [12]. The solution to this was simply to observe the general trend of decreasing errors of the testing data and terminate training when the testing data error flattened out or started decreasing.

TABLE II
VPPAW DATA USED TO TRAIN AND TEST BACK PROPAGATION NETWORKS FOR WELD MODELING AND EQUIPMENT PARAMETER SELECTION. WELDS NO. 2, 7, AND 12 (HIGHLIGHTED IN THE TABLE) WERE NOT USED FOR NETWORK TRAINING, BUT USED EXCLUSIVELY FOR TESTING.

Weld No.	Torch Standoff [mm]	Forward Current [Amps]	Reverse Current [Amps]	Travel Speed [mmps]	Crown Width [mm]	Root Width [mm]
1	4.0	136.7	120.6	3.40	9.50	7.11
2	**2.0**	**133.1**	**136.7**	**2.13**	**8.89**	**6.60**
3	4.0	132.7	134.3	2.13	10.16	6.86
4	8.0	129.1	135.7	2.12	13.21	15.49
5	2.0	132.3	136.9	4.25	7.37	4.57
6	4.0	133.1	133.7	4.24	8.89	5.59
7	**8.0**	**129.1**	**131.5**	**4.25**	**11.94**	**10.92**
8	2.0	131.5	137.1	5.52	6.86	4.57
9	4.0	130.6	136.8	5.53	8.13	5.59
10	8.0	131.3	127.6	5.53	9.65	6.86
11	2.0	131.3	137.0	6.79	6.60	4.32
12	**4.0**	**131.0**	**132.1**	**6.79**	**8.13**	**5.08**
13	8.0	134.2	130.7	6.79	9.14	6.86

Other network structures were tested, in addition to the one discussed above. Networks using a single layer of 5 to 20 nodes were tested, with only minor performance differences. Once the networks had been trained adequately (with on the order of 10,000 to 15,000 iterations), the worst case errors differed from one network to another by less than 1.0%.

The results of the tests are summarized in Table III. Again, note that the testing data are boldfaced in the Table, and these welds were not used for training the network. Although the errors in bead width estimates occasionally exceed 20% (in less than 8% of the outputs), the width modeling accuracy is typically on the order of 10% or less. Furthermore, it should be noticed that the worst performance is obtained in weld No. 4, in which the root width is recorded as <u>larger</u> than the crown width, which is an anomalous

condition with respect to all other welds. Finally, as shown here, the worst case error for the testing-only welds was 7.9%.

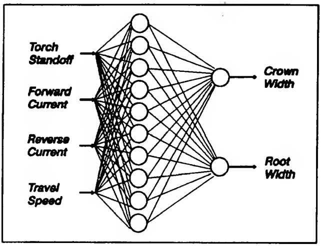

Fig. 4. The neural network model of the VPPAW process.

TABLE III
MODELING RESULTS, USING A NEURAL NETWORK OF A SINGLE HIDDEN LAYER WITH 10 NODES. THE NETWORK WAS TRAINED WITH THE ENTRIES WHICH ARE NOT BOLDFACED, WHILE THE BOLDFACED ENTRIES WERE USED EXCLUSIVELY FOR TESTING.

Weld No.	Actual Crown Width [mm]	Actual Root Width [mm]	Network Crown Width [mm]	Network Root Width [mm]	Crown Width Error [%]	Root Width Error [%]
1	9.50	7.11	9.09	6.74	-4.3	-5.2
2	**8.89**	**6.60**	**8.81**	**6.45**	**-0.9**	**-2.3**
3	10.16	6.86	9.95	8.12	-2.1	18.3
4	13.21	15.49	11.99	11.53	-9.3	-25.6
5	7.37	4.57	7.77	5.01	5.4	9.6
6	8.89	5.59	8.67	6.23	-2.5	11.5
7	**11.94**	**10.92**	**11.15**	**10.06**	**-6.6**	**-7.9**
8	6.86	4.57	7.28	4.36	6.1	-4.5
9	8.13	5.59	8.16	5.55	0.3	-0.6
10	9.65	6.86	10.32	8.68	6.9	26.5
11	6.60	4.32	6.84	3.80	3.7	-12.1
12	**8.13**	**5.08**	7.61	**4.80**	**-6.4**	**-5.5**
13	9.14	6.86	9.14	6.89	0.0	0.4

The conclusion from these modeling experiments is therefore, that the tested network appears to constitute a workable model for the tested VPPAW parameters.

V. VPPAW PROCESS CONTROL

The process control problem, as defined here, is to determine the required equipment parameters (the IWP) required to realize a set of desired output parameters (the DWP). To test the ability of a neural network to do this, a network was constructed to determine the torch standoff, forward current, reverse current, and travel speed for desired crown width and root width (refer to Fig. 5). Again, a single layer of 10 nodes was used, and the same thirteen welds were used as for the modeling experiment. For the parameter selecting network, however, the inputs were two (root width and crown width) and the outputs were four (torch standoff, forward current, reverse current, and travel speed). The same ten welds, as used for training the modeling network, were used for training this network. The network was trained through 15,000 iterations, resulting in sets of suggested torch standoff, forward current, reverse current, and travel speed selections. The network selected these parameters for each of the training welds, as well as for those which were not used in the training process.

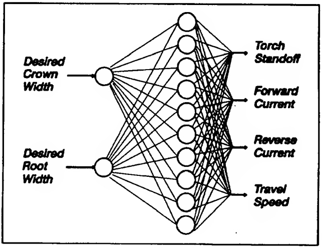

Fig. 5 A neural network used for VPPA welding equipment parameter selection.

To test the equipment parameter selection network shown in Fig. 5, its output was fed into the VPPA weld modeling network discussed previously (refer to Fig. 6). The desired crown and root widths are specified and applied to the inputs of the equipment parameter selection network which, in turn, determines the suitable torch standoff, forward and reverse current, and travel speed. As a substitute for the actual welding process, the neural network VPPA weld model, discussed earlier, is fed with these equipment parameters and its crown and root widths are compared with the desired ones. The results of this experiment are summarized in Table IV.

As shown in Table IV, the deviations in the ultimate crown and root widths from the desired ones are mostly very small. The worst case is again for weld 4 (root width error: -26.5%), where the root width was recorded as larger than the crown width. Generally the welding data which was left out

of the training of the parameter selector and the model (welds No. 2, 7, and 12) does not exhibit noticeably worse performance than the training data. The conclusions from this experiment suggest that neural networks may be reliably used in selecting welding equipment parameters for the VPPAW process.

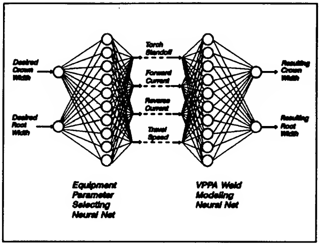

Fig. 6. A parameter selection network determines the welding equipment parameters necessary to achieve desired bead geometry. These equipment parameters can be used in the actual VPPAW process or, as illustrated here, in a weld model for demonstration purposes.

TABLE IV
COMPARISON OF CROWN AND ROOT WIDTH OBTAINED FROM A WELD MODEL AND THE CORRESPONDING WIDTHS SPECIFIED FOR THE EQUIPMENT PARAMETER SELECTING NETWORK

Weld No.	Desired Crown Width [mm]	Desired Root Width [mm]	Model Crown Width [mm]	Model Root Width [mm]	Crown Width Error [%]	Root Width Error [%]
1	9.50	7.11	9.47	7.39	-0.3	4.0
2	**8.89**	**6.60**	**8.95**	**6.64**	**0.7**	**0.6**
3	10.16	6.86	9.79	7.87	-3.7	14.7
4	13.21	15.49	11.93	11.38	-9.7	-26.5
5	7.37	4.57	7.62	4.80	3.4	5.0
6	8.89	5.59	8.68	6.25	-2.4	11.8
7	**11.94**	**10.92**	**11.32**	**10.32**	**-5.2**	**-5.5**
8	6.86	4.57	7.41	4.52	8.0	-1.0
9	8.13	5.59	8.23	5.63	1.3	0.7
10	9.65	6.86	9.49	7.42	-1.7	8.1
11	6.60	4.32	7.27	4.34	10.1	0.5
12	**8.13**	**5.08**	**8.11**	**5.46**	**-0.3**	**7.5**
13	9.14	6.86	9.34	7.20	2.1	4.9

VI. WELD PROFILE ANALYSIS AND CONTROL

A welding seam tracker, based on a laser scanner, is used at the NASA Marshall Space Flight Center for scanning VPPA weld profiles in near-real time. The seam tracking system is shown schematically in Fig. 7.

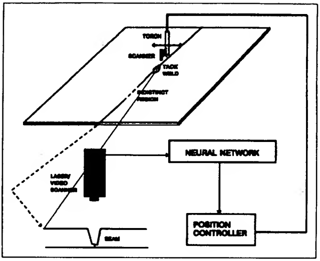

Fig. 7. The seam tracking system.

The objective of the seam tracker is to monitor the location of the unwelded seam with respect to the moving welding torch, and to adjust the location of the torch with respect to the seam in real-time. In addition to seam tracking applications, the profile scanning system can be utilized to evaluate the quality of the VPPA weld. The quality of a VPPAW root pass can, to a large extent, be determined from the shape of its surface profile. Therefore, means for automatically detecting abnormal shapes, excessive undercuts, non symmetry of the weld profile, etc., are important.

Referring to Fig. 7, the laser scanner scans the seam in front of the moving torch and locates the seam with respect to the torch. Although the laser signal produces an adequate indication of this location most of the time, there are occasions when this method may become unreliable or fail. One is where isolated tack welding points along the welding seam cover the seam line, resulting in a laser signal that confuses the tracking system. Another cause for tracking errors occurs when the junction between the joined parts (e.g., for butt welds) is very faint, or blends in with the parent metal surfaces, so that the seam apparently disappears over a section of the joint. Previously used data analysis algorithms for processing the output of the laser scanner sometimes became unreliable when the signal indicating the seam location became degraded in the manner discussed. The ability of the neural network to ignore minor disturbances made it an ideal candidate for this purpose.

A weld bead profile is obtained from the laser scanner, where it is available as a list of coordinates. Typically, the list contains on the order of 80 coordinates, obtained for a fixed cross section of the weld profile. Each coordinate consists of the location of the measured point, as measured transversely across the bead width (y-axis value) and the

height of the bead at that location (z-axis value). To locate, for example, the crown, undercuts and parent metal boundaries of the bead, the entire 80 data point heights (z-values) are fed into the network as distinctive inputs (refer to Fig. 8). Based on this given profile the network determines the locations of the desired bead profile parameters and presents these values as 5 separate outputs. Given the locations of the crown, undercuts, and bead boundaries, the crown height and undercut levels are easily determined by looking up the z-axis values at these locations. Deriving other properties of the bead, such as symmetry of the undercuts, etc., is straightforward once the above information has been obtained.

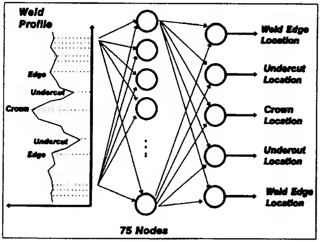

Fig. 8. The neural network system used to locate the crown, undercuts and weld boundaries from weld profile data.

Referring to Fig. 8, the height values of the digitized profiles were entered into the 60 inputs of the neural network, which was trained to locate the five features of the profile and present these locations at the outputs. Fig. 9 illustrates a typical cross sectional weld profile. The objective of the neural network was to find the locations of the weld crown, the left and right undercut, and the left and right weld boundaries. To the human observer, these features are relatively clear, with the possible exception of the weld boundaries. The locations of these features, as determined by the neural network, are indicated on the Figure with the arrows marked "N". For comparison, the previously used data analysis algorithms located the same features as shown with the arrows pointing from below the profile trace. In this case the previous method failed in correctly identifying the right weld boundary, while the neural network succeeded. In most cases the previous system performed satisfactorily, while occasionally it gave incorrect results. The errors usually occurred in determining the weld boundaries, and these errors could be significant. For comparison, the neural network performed somewhat better, on the average, and the worst case errors were always much smaller than those of the previous system.

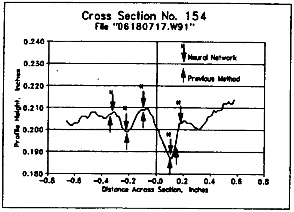

Fig. 9. A scanned weld profile, with the weld crown, undercuts, and parent metal boundaries indicated by the proposed neural network and the previously used system.

A total of 60 laser bead profiles were used to test and train the neural network used in the weld profile analysis. Of the total of 60 profiles, 30 were used for training and 30 were used for testing the trained network. Six of the testing profiles were sampled for error analysis, and the results are listed in Table V. As shown there, the errors are typically less than 1%.

TABLE V
SAMPLE RESULTS OBTAINED WITH THE WELD PROFILE ANALYSIS NEURAL NETWORK. SIX PROFILES SHOWN WERE NOT USED IN TRAINING THE NETWORK.

Profile:	L-Par Err:	L-Ucut Err:	Crown Err:	R-Ucut Err:	R-Par Err:	RMS Err:
134.17	-0.5%	0.6%	-0.4%	-0.2%	1.8%	0.9%
148.17	0.6%	0.7%	-0.3%	-0.1%	-1.1%	0.7%
302.17	0.2%	0.6%	1.0%	0.8%	0.0%	0.6%
316.17	0.4%	-0.2%	0.3%	0.1%	0.3%	0.3%
446.17	1.2%	-0.1%	-0.3%	0.2%	0.4%	0.6%
453.17	-0.7%	-0.1%	-0.2%	-0.7%	-0.6%	0.5%

VII. CONCLUSIONS

The experiments and analysis carried out for this work confirms that artificial neural networks are powerful tools for analysis, modeling, and control applications. They are particularly attractive in view of their capabilities to process nonlinear and noisy data, learn from actual welding data, and execute at relatively high speed. All of the areas examined during this research appear to have high potential for practical applications.

It was shown in this work that neural networks are capable of modeling parameters of the VPPAW process to on the order of 10% accuracy or better. The same was observed

when neural networks were used to select welding equipment parameters and the resulting bead geometries were estimated. These performance figures suggest that a VPPA welding control system can be implemented based on neural network models and control mechanisms.

The results obtained with analyzing weld profile data suggest that artificial neural networks can yield real-time results of equal or better accuracy and reliability than previously used data analysis algorithms. A neural network system can therefore be proposed for real-time as well as off-line quality control, based on observations of the weld bead profile. Furthermore, a neural network-based system can be used to enact corrective actions on the system variables that affect the bead profile. For example, a neural network system could be implemented to maintain seam tracking or torch orientation so that the weld bead would be maintained symmetrically on the center of the welded seam.

ACKNOWLEDGMENTS

This work has been partially funded by NASA contracts NAS8-37401 and NAS8-38918. Acknowledgments are due to Clyde S. Jones, Arthur C. Nunes, and Kirby G. Lawless at the NASA Marshall Space Flight Center for their support of this work.

REFERENCES

[1] R. L. O'Brien, *Welding Handbook - Welding Processes*, 8th ed., vol. 2, Miami: American Welding Society, 1991.

[2] G. E. Cook, K. Andersen, and R. J. Barnett, "Welding and Bonding," in *The Electrical Engineering Handbook*, R. C. Dorf, Ed., Boca Raton: CRC Press, 1993, pp. 2223-2237.

[3] G. E. Cook, K. Andersen, G. Karsai, and K. Ramaswamy, "Artificial Neural Networks Applied to Arc Welding Process Modeling and Control," *IEEE Transactions on Industry Applications*, vol. 26, no. 5, Sept./Oct. 1990, pp. 824-830.

[4] K. Andersen, *Synchronous Weld Pool Oscillation for Monitoring and Control*, Ph.D. Dissertation, Vanderbilt University, May 1993.

[5] A. C. Nunes, Jr., E. O. Bayless, Jr., C. S. Jones, III, P. M. Munafo, A. P. Biddle, and W. A. Wilson, "Variable Polarity Plasma Arc Welding On the Space Shuttle External Tank," *Welding Journal*, vol. 63, no. 9, pp. 27-35, Sept. 1984.

[6] M. Tomsic and S. Barhorst, "Keyhole Plasma Arc Welding of Aluminum with Variable Polarity Power," *Welding Journal*, vol. 63, no. 2, pp. 25-32, Feb. 1984.

[7] G. E. Cook, "Feedback and Adaptive Control in Automated Arc Welding Systems," *Metal Construction*, vol. 13, no. 9, pp. 551-556, Sept. 1981.

[8] R. J. Barnett, *Sensor Development for Multi-Parameter Control of Gas Tungsten Arc Welding*, Ph.D. Dissertation, Vanderbilt University, May 1993.

[9] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, pp. 4-22, April 1987.

[10] K. Andersen, R. J. Barnett, J. F. Springfield, and G. E. Cook, *Welding Technology: VPPAW Modeling and Control Using Artificial Neural Networks*, SBIR Phase I Final Report, NASA Contract No. NAS8-38918, Aug. 1991.

[11] G. E. Cook, K. Andersen, and R. J. Barnett, "Feedback and Adaptive Control in Welding," Recent Trends in Welding Science and Technology, S. A. David and J. M. Vitek, Eds., Materials Park: ASM International, 1990, pp. 891-903.

[12] K. Andersen, *Studies and Implementation of Stationary Models of the Gas Tungsten Arc Welding Process*, M.S. Thesis, Vanderbilt University, May 1992.